

WHAT IS CLAIMED IS:

1 1. A method implemented in circuitry, comprising:
2 accessing a program comprising a plurality of instructions including at least one
3 no operation (NOP) instruction; and
4 removing at least one NOP instruction in the program that is not needed to
5 provide a processing delay to ensure data is available to at least one dependent instruction
6 accessing the data.

1 2. The method of claim 1, wherein removing the at least one NOP instruction
2 comprises at least one of:
3 deleting one NOP instruction in the program that is not needed to provide the
4 processing delay to ensure the data is available to at least one dependent instruction
5 without moving a non-NOP instruction; and
6 replacing one NOP instruction with one determined instruction preceding the
7 NOP instruction in response to determining that one instruction preceding at least one
8 NOP instruction is capable of replacing the NOP instruction in the program.

1 3. The method of claim 1, further comprising:
2 deleting at least one instruction in the program that is not needed to provide the
3 processing delay to ensure the data is available to at least one dependent instruction; and
4 after deleting the at least one instruction, replacing at least one NOP instruction
5 with one determined instruction preceding the at least one NOP instruction in response to
6 determining that one instruction preceding at least one NOP instruction is capable of
7 replacing the NOP instruction in the program.

1 4. The method of claim 3, further comprising:
2 performing an additional iteration of deleting at least one instruction and then
3 replacing the at least one NOP instruction in response to replacing at least one NOP
4 instruction.

1 5. The method of claim 1, wherein the instructions in the program comprise
2 assembly language instructions coded by a developer.

1 6. The method of claim 1, wherein removing NOP instructions in the
2 program further comprises accessing and processing each NOP instruction by:
3 determining whether the accessed NOP instruction is needed to delay processing
4 of one dependent instruction following the accessed NOP instruction to ensure that data
5 is available to the dependent instruction accessing the data; and
6 deleting the accessed NOP instruction in response to determining that the NOP
7 instruction is not needed to ensure that data is available to the dependent instruction
8 accessing the data..

1 7. The method of claim 6, wherein determining whether the accessed NOP
2 instruction is needed to delay processing of one dependent instruction further comprises:
3 identifying instructions preceding the NOP instruction that have a delay in writing
4 the results. and
5 identifying dependent instructions following the NOP instruction that are
6 dependent on an availability of data from the identified instructions having the delay in
7 writing the results.

1 8. The method of claim 1, wherein removing NOP instructions in the
2 program further comprises accessing and processing each NOP instruction by:
3 replacing the accessed NOP instruction with one previous non-NOP instruction
4 that is capable of being moved forward to replace the accessed NOP instruction without
5 preventing data from being available to one dependent instruction following the NOP
6 instruction.

1 9. The method of claim 8, wherein the one previous instruction comprises a
2 preceding instruction closest to the accessed NOP instruction in the program.

1 10. The method of claim 8, wherein removing the NOP instructions further
2 comprises:
3 deleting at least one NOP instruction not needed to ensure that data accessed by
4 the dependent instruction is available to the dependent instruction, wherein the operations
5 of replacing accessed NOP instructions with previous non-NOP instructions are
6 performed after deleting NOP instructions not needed to ensure that data accessed by the
7 dependent instruction is available.

1 11. The method of claim 1, wherein removing NOP instructions in the
2 program further comprises accessing each NOP instruction and performing:
3 replacing the accessed NOP instruction with one previous non-NOP instruction
4 that is capable of being moved forward to replace the accessed NOP instruction without
5 preventing data from being available to one dependent instruction following the NOP
6 instruction and that is not a branch target instruction.

1 12. The method of claim 1, wherein the program instructions are for execution
2 by an engine in a multiprocessor engine.

1 13. A system for processing a plurality of instructions including at least one
2 no operation (NOP) instruction, comprising:
3 circuitry operable to:
4 access the program; and
5 remove at least one NOP instruction in the program that is not needed to provide a
6 processing delay to ensure data is available to at least one dependent instruction
7 accessing the data.

1 14. The system of claim 13, wherein the operation to remove the at least one
2 NOP instruction is further operable to:
3 delete one NOP instruction in the program that is not needed to provide the
4 processing delay to ensure the data is available to at least one dependent instruction
5 without moving a non-NOP instruction; and

6 replace one NOP instruction with one determined instruction preceding the NOP
7 instruction in response to determining that one instruction preceding at least one NOP
8 instruction is capable of replacing the NOP instruction in the program.

1 15. The system of claim 13, wherein the circuitry is further operable to:
2 delete at least one instruction in the program that is not needed to provide the
3 processing delay to ensure the data is available to at least one dependent instruction; and
4 after deleting the at least one instruction, replace at least one NOP instruction with
5 one determined instruction preceding the at least one NOP instruction in response to
6 determining that one instruction preceding at least one NOP instruction is capable of
7 replacing the NOP instruction in the program.

1 16. The system of claim 15, wherein the circuitry is further operable to:
2 perform an additional iteration of deleting at least one instruction and then
3 replacing the at least one NOP instruction in response to replacing at least one NOP
4 instruction.

1 17. The system of claim 13, wherein the instructions in the program comprise
2 assembly language instructions coded by a developer.

1 18. The system of claim 13, wherein the operation to remove NOP
2 instructions in the program further comprises accessing and processing each NOP
3 instruction to:
4 determine whether the accessed NOP instruction is needed to delay processing of
5 one dependent instruction following the accessed NOP instruction to ensure that data is
6 available to the dependent instruction accessing the data; and
7 delete the accessed NOP instruction in response to determining that the NOP
8 instruction is not needed to ensure that data is available to the dependent instruction
9 accessing the data.

1 19. The system of claim 18, wherein the operation to determine whether the
2 accessed NOP instruction is needed to delay processing of one dependent instruction is
3 further operable to:

4 identify instructions preceding the NOP instruction that have a delay in writing
5 the results. and

6 identify dependent instructions following the NOP instruction that are dependent
7 on an availability of data from the identified instructions having the delay in writing the
8 results.

1 20. The system of claim 13, wherein the operation to remove NOP
2 instructions in the program further accesses and processes each NOP instruction to:

3 replace the accessed NOP instruction with one previous non-NOP instruction that
4 is capable of being moved forward to replace the accessed NOP instruction without
5 preventing data from being available to one dependent instruction following the NOP
6 instruction.

1 21. The system of claim 21, wherein the one previous instruction comprises a
2 preceding instruction closest to the accessed NOP instruction in the program.

1 22. The system of claim 13, wherein the operation to remove the NOP
2 instructions is further operable to:

3 delete at least one NOP instruction not needed to ensure that data accessed by the
4 dependent instruction is available to the dependent instruction, wherein the operations of
5 replacing accessed NOP instructions with previous non-NOP instructions are performed
6 after deleting NOP instructions not needed to ensure that data accessed by the dependent
7 instruction is available.

1 23. The system of claim 13, wherein the operation to remove NOP
2 instructions in the program further accesses each NOP instruction to:

3 replace the accessed NOP instruction with one previous non-NOP instruction that
4 is capable of being moved forward to replace the accessed NOP instruction without

5 preventing data from being available to one dependent instruction following the NOP
6 instruction and that is not a branch target instruction.

1 24. An article of manufacture enabled to:
2 access a program comprising a plurality of instructions including at least one no
3 operation (NOP) instruction; and
4 remove at least one NOP instruction in the program that is not needed to provide a
5 processing delay to ensure data is available to at least one dependent instruction
6 accessing the data.

1 25. The article of manufacture of claim 24, wherein the operation to remove
2 the at least one NOP instruction is further operable to:
3 delete one NOP instruction in the program that is not needed to provide the
4 processing delay to ensure the data is available to at least one dependent instruction
5 without moving a non-NOP instruction; and
6 replace one NOP instruction with one determined instruction preceding the NOP
7 instruction in response to determining that one instruction preceding at least one NOP
8 instruction is capable of replacing the NOP instruction in the program.

1 26. The article of manufacture of claim 24, wherein the article of manufacture
2 is further operable to:
3 delete at least one instruction in the program that is not needed to provide the
4 processing delay to ensure the data is available to at least one dependent instruction; and
5 after deleting the at least one instruction, replace at least one NOP instruction with
6 one determined instruction preceding the at least one NOP instruction in response to
7 determining that one instruction preceding at least one NOP instruction is capable of
8 replacing the NOP instruction in the program.

1 27 The article of manufacture of claim 26, wherein the article of manufacture
2 is further operable to:

3 perform an additional iteration of deleting at least one instruction and then
4 replacing the at least one NOP instruction in response to replacing at least one NOP
5 instruction.

1 28. The article of manufacture of claim 24, wherein the instructions in the
2 program comprise assembly language instructions coded by a developer.

1 29. The article of manufacture of claim 24, wherein the operation to remove
2 NOP instructions in the program further accesses and processes each NOP instruction to:
3 determine whether the accessed NOP instruction is needed to delay processing of
4 one dependent instruction following the accessed NOP instruction to ensure that data is
5 available to the dependent instruction accessing the data; and
6 delete the accessed NOP instruction in response to determining that the NOP
7 instruction is not needed to ensure that data is available to the dependent instruction
8 accessing the data.

1 30. The article of manufacture of claim 29, wherein the operation to determine
2 whether the accessed NOP instruction is needed to delay processing of one dependent
3 instruction is further operable to:
4 identify instructions preceding the NOP instruction that have a delay in writing
5 the results. and
6 identify dependent instructions following the NOP instruction that are dependent
7 on a availability of data from the identified instructions having the delay in writing the
8 results.

1 31. The article of manufacture of claim 24, wherein the operation to remove
2 NOP instructions in the program further accesses and processes each NOP instruction to:
3 replace the accessed NOP instruction with one previous non-NOP instruction that
4 is capable of being moved forward to replace the accessed NOP instruction without
5 preventing data from being available to one dependent instruction following the NOP
6 instruction.

1 32. The article of manufacture of claim 31, wherein the one previous
2 instruction comprises a preceding instruction closest to the accessed NOP instruction in
3 the program.

1 33. The article of manufacture of claim 31, wherein the operation to remove
2 the NOP instructions is further operable to:
3 delete at least one NOP instruction not needed to ensure that data accessed by the
4 dependent instruction is available to the dependent instruction, wherein the operations of
5 replacing accessed NOP instructions with previous non-NOP instructions are performed
6 after deleting NOP instructions not needed to ensure that data accessed by the dependent
7 instruction is available.

1 34. The article of manufacture of claim 24, wherein the operation to remove
2 NOP instructions in the program further accesses each NOP instruction to:
3 replace the accessed NOP instruction with one previous non-NOP instruction that
4 is capable of being moved forward to replace the accessed NOP instruction without
5 preventing data from being available to one dependent instruction following the NOP
6 instruction and that is not a branch target instruction.

1 35. The article of manufacture of claim 24, wherein the program instructions
2 are for execution by an engine in a multiprocessor engine.